

REMARKS:

With regard to the Examiner's claim rejections under 35 U.S.C. Article 103, Examiner rejects Claims 1, 14 and 28 as being obvious when *Phanouriou et al* is combined with *Eagar et al* and *Apte et al*. Moreover, the Examiner contends that the remaining dependent claims are obvious as they only teach features described in *Phanouriou*.

Applicant has amended each of the Claims 1, 14 and 28.

The claimed invention, in at least the independent claims, clearly teaches that the defined invention automatically translates legacy source code interface specification definitions into an executable software component which may be executed on an interconnected network computing resource.

Support for this feature can be found at many sections of the specification. For example, at Page 15, Lines 23 to 25 it is stated:

"... the automated system for creating the client side application (20) will hereinafter be known as ActiveLINC".

Also at Page 16, Line 32 through to Page 17, Line 6 it is stated (in part):

"... by creating the active link components 22... client programmers can choose either to write their own client applications, web pages and scripts... or to make use of the user interface applications 21 that can be generated automatically from their existing terminal screen and GUI form definitions".

As stated clearly in previous responses, *Phanouriou et al.* ("Transforming Command Line Driven Systems To Web Applications") teaches a software application that allows a user or a programmer to manually create a graphical user interface for a legacy application that utilises a command driven interface.

The Examiner's attention is drawn to Page 6 of the *Phanouriou* disclosure, which clearly states that the graphical user interface (GUI) produced by the software application disclosed in *Phanouriou* is generated "from an application description and a set of UI mapping rules, generated by a person familiar with how to run the application".

As submitted to the Examiner in a number of previous responses, the presently claimed invention automatically generates the required executable software components by utilizing the original legacy code. That is, the present invention allows an interface to be built from original legacy code, without the need for any substantive input from a programmer or user!

Therefore, it is clear that *Phanouriou* has no teaching of automatically generating executable source code.

Turning to *Eagar*, even if *Eagar* were to be combined with *Phanouriou*, there is no teaching in *Eagar* of automatically generating software applications based on original legacy source code.

In the Office Action, Examiner refers to *Eagar* Column 23, Lines 35 to 45. The paragraph referred to by the Examiner does not teach or imply the automatic generation of software components.

The *Eagar* phrase "information stored in the user interface repository is converted at application

runtime into the user interface representation structures of the presentation layer 116" merely describes a process by which any GUI operates.

That is, a series of parameters are stored in a file, and these parameters are called upon by a GUI application when a user command requires the GUI to be rendered.

The term "user interface repository" of Eagar clearly denotes a structure which determines the layout of the interface. Rather, it is merely a file that contains information which is used to render particular features of a GUI. This is not legacy code.

Even if the Examiner were to interpret this feature as being "legacy code", the Examiner's attention is drawn to Eagar Line 66 in Column 23 through to Line 11 in Column 24. At this section of the specification, Eagar teaches that:

"... GUI files 248 are used by application developers and maintenance personnel to modify application screens and messages as part of the re-engineering system 30".

In other words, any substantive changes to the graphical user interface are performed manually by programmers!

Turning to Apte, there is also absolutely no disclosure in Apte of automatically generating a user interface derived from legacy source code, wherein the generated user interface may be run on a network computing device.

Therefore, even if all three applications are combined, at least one or more features of the presently claimed invention is not taught. For example, Applicant's method teaches:

(i) building an interface from original legacy code without need for special input from a programmer or user;

(ii) automatically generating executable source code where Applicant teaches "automatic" translation of legacy source code interface specification definitions into an executable software component which can be executed on a network computing resource.

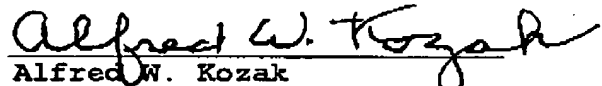
For this reason, the obviousness objection raised by the Examiner cannot be considered sustainable.

Attached hereto is Appendix I which is used to summarize the technological and the legal issues in accordance with the present status of Patent Law and current Court decisions.

As a result, it is respectfully requested that Examiner consider Applicant's claims as a whole in their entirety and subsequently provide a timely Notice of Allowance.

Respectfully submitted,

BY


Alfred W. Kozak
Reg No. 24,265
(949) 380-5822

CERTIFICATE OF MAILING / TRANSMISSION (37 CFR 1.84)

I hereby certify that this correspondence is, on the date shown below, being:

MAILING

☐ deposited with the United States Postal Service as first class mail in an envelope addressed to:
Commissioner for Patents, PO Box 1450,
Alexandria, VA 22313-1450.

Date January 9, 2007

FACSIMILE

☒ transmitted by facsimile to the United States Patent and Trademark Office.

 January 9, 2007
Carol A. Wasserman Date

Appendix 1

In order to reiterate both the legal factors and the technical factors involved in Applicant's system, Applicants are including this Appendix I to summarize the issues involved.

Examiner has rejected claims 1-7, 10-20, 23-26, 27-34 and 37-40 for obviousness under 35 USC 103(a), as unpatentable over Phanouriou in view of Eager, U.S. Patent 5,960,200.

While Examiner argues that Phanouriou teaches the substance of these claims, then Examiner indicates that Phanouriou does not explicitly teach providing a software application which utilizes the legacy source code to "automatically" produce a series of executable components that provide the functionality for interaction with the legacy software application.

Then, Examiner says that the Eager reference, U.S. Patent 5,960,200, provides a software application which utilizes the legacy source code to "automatically" produce a series of executable software components that provide the functionality for interaction with the legacy software application, citing structure 118; column 23, lines 35-45 and column 9, lines 45-65.

Here it is to be noted, that the Eager Fig. 4 indicates the structure 118 as a display platform user interface representation structure. Further, the citation to Eager column 9, lines 45-65, and column 23 lines 35-45 ----- on any clear reading and interpretation, do certainly not indicate any automatic production of executable software components from legacy source code.

Examiner argues that the Eager technology could be combined with the Phanouriou technology to constitute the substance of Applicants' invention.

Applicants would herein traverse any such conclusion on the part of the Examiner.

Further, the Examiner's contention that the Eager reference technology could be combined with the technology of Phanouriou, will be seen to be a rather idealistic presumption.

Phanouriou makes use of a tool designated Javamatic, which can create user interfaces in JAVA. Phanouriou shows a method to add a Web-based interface to a command-line driven application without programming. The method uses a "high-level" description of an application to automatically generate a user interface (UI) and then invokes commands in the legacy application transparently. It does not require any changes to the application code, nor does it require application recompilation with special toolkits. The application can be written in any programming language as long as the needed functionality is accessible from the command line.

Javamatic (in Phanouriou) can transform legacy command-line driven applications to Web applications. The legacy code sits on a remote machine and is wrapped with a Javamatic interface server. The user interface is generated on-the-fly by a Javamatic interface client on the user's www browser.

Thus, the Phanouriou system involves a very specialized kind of architecture and operation. But Eager involves a different architecture and operations to provide a system to transition an entire business enterprise to a distributed

infrastructure. The infrastructure is a multi-tiered client/server target architecture that adheres to open system standards. The multi-tiered architecture includes four specialized layers. Each legacy (source) application must be identified and prioritized.

The Eager system has automated capability to translate existing source applications into new target applications on a multi-tiered client/server architecture. Such translation requires conversion of user interfaces, procedural languages and data definitions. A common intermediate language is required.

Note the complexity of the Eager configuration compared to Applicant's arrangement of Fig. 1 which can be seen to be much simpler and less sophisticated, and less prone to error.

Now, the question arises --- how could the Eager technology be assimilated, implemented, or added into the technology of Phanouriou?

The answer here is --- it could not, since these are two completely different operating systems and technologies which are not compatible with each other, and for the Examiner to contend that this would just be a simple matter of combining one technology to another technology --- would be beyond the comprehension of the average engineer.

Additionally, it should be reiterated that the cited portions of Eager, namely item #118, and the two cited paragraphs which Examiner enumerated, --- these do not indicate any "automatic" operation whereby a software application utilizing the legacy source code can automatically produce a series of executable software components that provide the functionality for interaction with a legacy software application.

In regard to the Eager reference, at column 2, under "Summary of the Invention" at lines 44-50, where Eager says --- his present invention includes the automated capability to translate existing source applications into new target applications on a multi-tiered client/server architecture, --- we must note the translation of source applications to target applications includes the conversion of user interfaces, procedural languages and data definitions. These conversions use a two-phase process where source program components written in the source languages are first translated to components in a common intermediate language. The intermediate language components are then translated to target program components in the target languages.

Here, Eager provides a specialized configuration of elements which is quite diverse and different from those discussed in the Phanouriou reference.

So again, the question arises, what group of Engineers would try to take the technology of Eager and adapt it into the technology of Phanouriou? There would be little or no likelihood of such a monumental engineering project and redesign to be accomplished.

Thus, when the Examiner indicates that it is just a matter of technology of Eager to be combined with the technology of Phanouriou, and that this would be the "equivalent" of Applicants' configuration and system ---- then this type of composite thinking on combining technologies certainly could not be deemed relevant to the technology and situation of Applicants' system.

As was previously mentioned, it is inappropriate, incorrect and improper for the Examiner to combine references by inserting a secondary reference into combination with a primary reference

on the basis that the two references could be compatibly and operably combined which are the equivalent of Applicants' configuration.

This is particularly true in the legal sense, since there is nothing in the first reference to Phanouriou to suggest that any such technology such as that of Eager, would be useful or could be combined into the Phanouriou reference.

Subsequently, Examiner has indicated that claims 8, 21, and 35 are rejected as obvious under 35 USC 103(a) as unpatentable over Phanouriou in view of Eager and further in view of Apte (U.S. Patent 6,662,236 B1).

Then later, Examiner rejects claims 9, 22 and 36 for obviousness under 35 USC 103(a) as unpatentable over Phanouriou, in view of Eager, in view of Harold. Applicants would traverse these conclusions.

Basically, in the present situation, the Examiner is basing his contentions by combining various technologies from various cited references and saying they could be combined to provide the same configuration that is stated in Applicants' claims. First, it must be said that these cited technologies could not provide the specific configuration of modules as provided by Applicants. And further, the present patent law requires that there should be indicated some "motivation" for an Examiner to conclude that the combination of such references is an operable possibility.

The Patent Office has set forth in MPEP 706.02 a three-step requirement for establishing a prima facie case of "obviousness". The first step requires that the Examiner must set forth the differences in the claim over the applied references.

The second step requires that the Examiner must set forth the proposed modification of the reference which would be necessary to arrive at the claimed subject matter. And, as a third step, this requires that the Examiner explain why the proposed modification (or combination) would be "obvious".

In the Court decision of *In re Bozek*, 133 USPQ 545 (CCPA 1969). Here, the Court held:

"One can properly rely . . . on a conclusion of obviousness from common knowledge and common sense of the person of ordinary skill in the art without any specific hint or suggestion in a particular reference."

But then specifically, in this case, the Courts required the Examiner to identify where the prior art provides a "motivating suggestion" for the modification, as for example, in the decision of *In re Jones*, 21 USPQ2d 1941 (Fed.Cir.1992), where the Court had held:

Before the PTO may combine the disclosures of two or more prior art references in order to establish prima facie obviousness there must be some suggestion for doing so. . . . *In re Fine*, 5USPQ2d 1596, 1598-99 (Fed.Cir.1988).

To add further to the legal thinking in this situation, a recent decision of the Board of Patent Appeals and Interferences (*In re Kahn*, Fed.Cir. No.04-1616, 3/22/06), the Board indicated that ---

an obviousness rejection must articulate the motivation, suggestion or teaching that would have led the skilled artisan at the time of invention to combine prior art elements to make the claimed invention. (underlines added).

This case of In re Kahn indicated that ---- absent such an explanation (reason for motivation) we infer, said the Court, that the Board used hindsight to conclude that the invention was obvious.

In this Kahn case, the Court said -- to establish a prima facie case of obviousness based on a combination of prior art elements, the Board must articulate the basis on which it concludes it would have been obvious to make the claimed invention ---- when the Board does not explain the motivation, the suggestion, or the teaching, ---- that would of lead the skilled artisan at the time of the invention to the claimed invention as a whole, we infer that the Board used hindsight to conclude that the invention was obvious.

Thus, it is incumbent upon the Examiner to present some evidence or reason why the Eager reference could or should be combined with the Phanouriou reference, and what exactly would be the reason or motivation for making such a combinative step?

As was previously mentioned in earlier amendments, the Courts have indicated that even if the prior art may be modified as suggested by the Examiner, the modification is not obvious unless the prior art suggests the desirability for the modification. As for example, in the decision of In re Fritch, 23 USPQ2d 1780 (Fed.Cir.1992), where the Court held:

Mere fact that prior art may be modified to reflect features of claimed invention does not make modification, and hence claimed invention obvious unless desirability of such modification is suggested by prior art (at page 1780).

Further, it has been later set forth that the "motivating suggestion" must be explicit as was decided in the case of Winner International Royalty Corp. v. Wang, Case No. 96-2107, 48 USPQ2d 1139 (District Court D.C. 1998), where the Court held:

Invention cannot be found obvious unless there was some explicit teaching or suggestion in art to motivate one of ordinary skill to combine elements so as to create same invention (underlines added).

In light of the above legal requirements, Applicants would ask --- where is the evidence or suggestion to motivate a skilled engineer to take the technology of Eager and apply it to the technology of Phanouriou?

Absent such evidence of motivation to do so, it is improper to contend that the Eager reference could so easily be applied to the Phanouriou reference to establish the equivalent of Applicants' configuration.

In view of the inappropriateness of combining the references in the manner that Examiner has tried to do, it is respectfully requested that the present claims be viewed as a whole in their entirety, and on the basis that neither of the references to Phanouriou, Eager, Apte or Harold, when put in combination, could provide the equivalency of Applicants' stated claims.